Manual de usuario del proyecto AI-Live

Índice de contenido

1 **Requisitos** 2
2 **Compilación** 2
2.1 Compilación recursiva 2
2.2 Compilación manual por componentes 2
2.3 Configuración de Make 2
3 **Configuración** 4
4 **Ejecución** 5
4.1 Ejecución utilizando script 5
4.1.1 Clientes 5
4.1.2 Servidor 5
4.1.3 Gui 5
4.1.4 Ayuda 6
4.2 Ejecución manual de los componentes 6

1 Requisitos

La aplicación está diseñada para cualquier sistema con una versión de compilador compatible con GCC y que utilice hilos o threads POSIX y sockets BSD por lo que solo es necesario un sistema con el compilador apropiado y las bibliotecas de desarrollo. Para la compilación en entornos windows se requiere el compilador MinGW correctamente instalado y utilizar su implementación de GNU Make.

Si se quieren utilizar el script de ejecución se necesita tener instalado xterm y un bash compatible.

Para utilizar el cliente gráfico 3d se necesita tener instalado ogre o en su defecto su SDK para compilar el proyecto con él y un interfaz de ventanas disponibles en el sistema para la visualización. El cliente gráfico no está aún portado para entornos windows.

2 Compilación

2.1 Compilación recursiva

Para compilar el proyecto basta con ejecutar "Make" en el directorio raíz y éste se encargará de compilar todos los módulos del sistema actual que incluyen servidor, cliente manual, cliente IA y la interfaz GUI 3d.
El código del motor clips, motor emocional y archivos comunes se compilan en el directorio "build" que ha de existir antes de iniciar la compilación.
Para limpiar los archivos generados hay un target en el archivo Makefile que se encarga de borrar los ejecutables compilados y los código objeto intermedios.

2.2 Compilación manual por componentes

Se han añadido opciones para compilar los distintos módulos de forma manual por separado.
Los módulos disponibles para compilar por separado son:

server compila el servidor
client compila el cliente IA
client-manual compila el cliente manual
gui compila la interfaz Gui 3d (Solo en Unix)

2.3 Configuración de Make

No es necesario editar el archivo Makefile para compilar el proyecto salvo si se quiere compilar en diferentes entornos. Para entornos Windows hay una línea específica para activar la correcta compilación bajo ese entorno, para el resto de plataformas Unix soportadas hay que comentar esa línea.
Si se desean cambiar los directorios u opciones de compilación, dentro del Makefile están declaradas las variables para ello. A continuación se muestra un fragmento de dicho Makefile y sus opciones:

```
#WINDOWS=1
```

```
OSTYPE=$(shell uname)

CC=gcc

CFLAGS = -Wall -lm


COMMON_DIR=common

CLIPS_DIR=$(COMMON_DIR)/clips

EENGINE_DIR=$(COMMON_DIR)/emotional_engine

SERVER_DIR=server-source

CLIENT_DIR=client-source

CLIENTM_DIR=clientmanual-source

GUI_DIR=client-gui

BUILD_DIR=build


COMMON_OBJS=$(BUILD_DIR)/buffer.o $(BUILD_DIR)/rinput.o

EENGINE_OBJS=$(BUILD_DIR)/ee_clips_adapter.o $(BUILD_DIR)/emotional_engine.o

SERVER_SRCS=$(SERVER_DIR)/server.c $(SERVER_DIR)/clientlist.c

CLIENT_SRCS=$(CLIENT_DIR)/client.c

CLIENTM_SRCS=$(CLIENTM_DIR)/client.c



ifndef WINDOWS

	GUI_DEFINES =

	GUI_LIBS = OGRE CEGUI-OGRE CEGUI OIS

	GUI_CXX = g++

	GUI_CXXFLAGS = $(shell pkg-config --cflags $(GUI_LIBS)) $(GUI_DEFINES)

	GUI_LD = g++

	GUI_LDFLAGS = $(shell pkg-config --libs $(GUI_LIBS))

	GUI_FINCLUDES = $(GUI_DIR)/serverCommunication.cpp
$(GUI_DIR)/auxiliarStringFunctions.cpp
```

```
endif
```

El directorio “BUILD_DIR” debe existir antes de iniciar la compilación.

3 Configuración

La configuración para ejecutar el proyecto se realiza en el fichero “conf-ialive” situado en el directorio raíz. A continuación se detallan las opciones disponibles

Variables globales para todos los clientes y servidor

rootdir=directorio raíz del proyecto, contiene los archivos necesarios para compilar y ejecutarlo
profilesdir= directorio donde se alojan todos los perfiles de personajes
serverPATH=directorio donde se encuentra el ejecutable del servidor
clientPATH=directorio del cliente
clientManualPATH=directorio del cliente manual
clientGUIPATH=directorio del cliente gráfico
ontology=nombre del fichero de ontología, debe estar en el directorio raíz de la aplicación.
Ontology_2=nombre del fichero de ontología para la habilidad, debe estar en el directorio raíz de la aplicación.
Ontology_3=nombre del fichero de ontología para la historia, debe estar en el directorio raíz de la aplicación.
port=puerto donde el servidor recibe peticiones
sizex=geometría x de la ventana xterm
sizey=geometría y de la ventana xterm
posx=incrementos x para colocar las venentas xterm
posy=incrementos y para colocar las ventanas xterm
cleanfiles=nombre del script para la limpieza de archivos
host=ip o hostname de la máquina a la que conectar los clientes

Variables para el servidor
initial_state=initial.state
server=./server

Variables para el cliente IA
client=ejecutable del cliente
stageC=escenario por defecto al que conectar

Variables del cliente manual
clientmanual=ejecutable del cliente manual
stageM=escenario por defecto al que conectar

Variables del cliente gráfico
clientgui=ejecutable del cliente gráfico

4 Ejecución

4.1 Ejecución utilizando script

Para lanzar el programa, tan sólo hay que ejecutar el script run.sh y unos parámetros determinados para lanzar los distintos módulos. Esta es la forma de uso y su explicación.

./run.sh -C [server|manual|client] -s -c [profile] -m [profile] -g

La opción -C sirve para limpiar trazas y ficheros de estadísticas de los clientes y servidor, tiene tres opciones, server client y manual que lanzan a su vez el script de limpieza de cada uno de esos módulos.

Ej: ./run.sh -C server -C client
Esta orden limpiaría las trazas y ficheros de estadísticas del cliente y los estados que se generan en el servidor.

4.1.1 Clientes

La opción **-c** (profile) lanza un cliente de IA utilizando el perfil que se indique. Se debe pasar el nombre del perfil sin la extensión propia ".profile" y se lee del directorio configurado anteriormente bajo la opción profilesdir

Ej: ./run.sh -c mike
Este ejemplo lanzaría un cliente de inteligencia artificial usando el perfil mike.profile que se debe encontrar bajo el directorio configurado en el archivo "conf-ialive" y con las opciones del mismo, para realizar la conexión al servidor y puerto predeterminados.

Opción: **-m** (profile) opción equivalente a la anterior pero para lanzar un cliente manual especificando el perfil a cargar de la misma forma.

Ej: ./run.sh -c mike -m amy
Este ejemplo ejecuta un cliente de inteligencia artificial autónomo utilizando el perfil mike.profile y un cliente manual con las características del perfil amy.profile

4.1.2 Servidor

Opción **-s** se utiliza para ejecutar un servidor con las opciones del archivo "conf-ialive" Nótese que solo puede haber un servidor escuchando en el mismo puerto de una máquina aunque podría haber varios servidores utilizando diferentes puertos de escucha.

4.1.3 Gui

Opción **-g** activa el cliente gráfico 3D con las opciones especificadas en su archivo gui.ini que está en el mismo directorio que el binario. Las opciones de ese archivo incluyen la ip o nombre del servidor al que conectarse, puerto y escenario que representará.

4.1.4 Ayuda

Opción **-h** muestra una pequeña ayuda con los parámetros disponibles. Si se ejecuta el script sin ningún parámetro también se muestra esta ayuda de uso.

4.2 Ejecución manual de los componentes

También se pueden ejecutar los componentes de forma manual ejecutando los binarios correspondientes y pasando a mano los parámetros necesarios. En entornos windows por el momento es la única forma de ejecutar los componentes:

Los clientes, tanto de IA como manual se lanzan de la siguiente manera

```
./client
Usage: client <ontology> <serverhost> <port> [<actor>] [<stage>]
(default stage is DEFAULT_STAGE, default actor is DEFAULT_ACTOR)
```

Los parámetros que se especifican en orden son la ruta a la ontología, ip o nombre del servidor, puerto al que conectarse y como opciones la ruta del perfil del actor a usar (de nuevo sin la extensión .profile) y el escenario al que conectarse.

Para lanzar el servidor manualmente se utilizan los parámetros: ruta de la ontología, estado inicial para generar el escenario y el puerto de escucha:

```
./server
Usage: server <ontology> <initial.state> <port>
```

Es importante conservar el orden de los parámetros como se especifica.